Bifurcation analysis of wind-driven flows with MOM4

Bernsen, Erik; Dijkstra, Henk A.; Wubs, Fred W.

Published in:
Ocean modelling

DOI:
10.1016/j.ocemod.2009.06.003

Publication date:
2009

Ocean Modelling 30 (2009) 95–105

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

Bifurcation analysis of wind-driven flows with MOM4

Erik Bernsen [a,*], Henk A. Dijkstra [a], Fred W. Wubs [b]

[a] Institute for Marine and Atmospheric Research Utrecht, Department of Physics and Astronomy, Utrecht University, 3584CC Utrecht, The Netherlands
[b] Department of Mathematics and Computer Science, University of Groningen, Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 9 November 2008
Received in revised form 28 May 2009
Accepted 11 June 2009
Available online 16 June 2009

Keywords:
Bifurcation analysis
Jacobian–Free Newton–Krylov method
Steady state
Wind-driven circulation
Nonlinear equations
Numerical analysis

ABSTRACT

In this paper, the methodology of bifurcation analysis is applied to the explicit time-stepping ocean model MOM4 using a Jacobian–Free Newton–Krylov (JFNK) approach. We in detail present the implementation of the JFNK method in MOM4 but restrict the preconditioning technique to the case for which the density distribution is prescribed. For a prescribed density field case, we present bifurcation diagrams, for the first time in MOM4, for the wind-driven ocean circulation. In addition, we show that the JFNK method can reduce the spin-up time to a steady equilibrium in MOM4 considerably if an accurate solution is required.

1. Introduction

Bifurcation analysis is a tool from dynamical systems theory to analyze transition phenomena in fairly general systems. As such it has been applied to quasi-geostrophic and shallow-water models of the wind-driven ocean circulation to determine multiple steady states in single- and double-gyre flows under steady wind forcing (Speich et al., 1995; Dijkstra and Katsman, 1997; Primeau, 2002). By changing parameters in these flows, the physics of transitions to more complex time-dependent flows could be analyzed (Nadiga and Luce, 2001; Simonnet et al., 2005). Techniques from bifurcation theory have also been used to study multiple equilibria of the Atlantic meridional overturning circulation which may occur when the freshwater flux on the northern North Atlantic is changed (Dijkstra and Weijer, 2005). The instabilities of these thermohaline ocean flows may lead to multidecadal variability which is likely associated with the Atlantic Multidecadal Oscillation (Raa et al., 2002).

To contrast the bifurcation analysis with transient ocean modeling let us write the ocean model equations, with state vector $\vec{x}$ (the prognostic variables at all grid points) as

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}, \lambda), \tag{1}$$

where λ is a certain (control) parameter in the model such as the wind-stress strength or the vertical diffusivity. In the transient approach using an explicit time-discretization scheme, a value of λ is chosen, an initial state is specified and the time evolution of $\vec{x}$ is calculated; this can be repeated for different values of λ. If one is in particular interested in how the equilibrium states (for $t \to \infty$) of the ocean flows depend on λ, then bifurcation analysis can be an efficient complementary tool. One important example is the determination of the so-called spin-up state which usually has to be obtained through extensive computation in a transient approach.

Instead of determining the solution of the model after a very large time, for example a steady state, in bifurcation analysis one aims to solve the equations

$$\vec{F}(\vec{x}, \lambda) = 0, \tag{2}$$

for $\vec{x}$ given a fixed λ. A Newton–Raphson type iterative method of the form

$$0 = \vec{F}(\vec{x}_n, \lambda) + J(\vec{x}_n)\delta\vec{x}, \tag{3}$$

is used most often, starting from a certain initial guess $\vec{x}_0$. In (3), $\delta\vec{x} = \vec{x}_{n+1} - \vec{x}_n$ and $J(\vec{x}_n)$ is the Jacobian matrix given by $J_{ij} = \partial F_i/\partial x_j$. To solve for $\delta\vec{x}$ in (3), a linear system of equations has to be solved.

Krylov subspace methods are very suitable for solving these (very) large and sparse linear systems. A prominent example of such a Krylov subspace method is the Generalized Minimum Residual (GMRES) method (Saad, 1996). For the linear systems derived from ocean model equations, these iterative methods only converge fast enough when a preconditioner is applied. The design of the preconditioner is the crucial problem in the application of bifurcation analysis to ocean models.

In de Niet et al. (2007), a bifurcation analysis of both wind- and buoyancy forced circulation was done with a primitive equation ocean model (THCM) which includes state-of-the-art mixing

* Corresponding author. Tel.: +31 30 2532978; fax: +31 30 2543163.
E-mail address: E.Bernsen@phys.uu.nl (E. Bernsen).

1463-5003/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ocemod.2009.06.003

parameterizations. The results shown in de Niet et al. (2007) were still for an idealized spherical basin configuration with low-resolution (4° horizontal resolution) but were recently extended to a high resolution with realistic continents using a parallel version of THCM (Thies, 2008). To perform the steady-state computations versus parameters with THCM, a tailored preconditioner was designed which was based on the availability of the Jacobian matrix and the structure of the governing primitive equations.

As it is more troublesome to derive explicit Jacobian matrices when more complex subgridscale parameterizations are added one would like to be able to perform bifurcation analysis on ocean models which were designed only for transient computations, such as POP, MOM and MICOM. Setting up the Newton–Raphson process using Krylov subspace methods is relatively easy because there is no need to construct the model's Jacobian explicitly. Only matrix–vector products with this Jacobian and certain vectors $\vec{v}$ are required and these can be determined (Knoll and Keyes, 2004) as

$$J(\vec{x})\vec{v} = \frac{\vec{F}(\vec{x} + \epsilon\vec{v}) - \vec{F}(\vec{x})}{\epsilon}, \tag{4}$$

for some small ϵ. When a Krylov subspace iteration is used, the solution technique is referred to as a Jacobian–Free Newton–Krylov (JFNK) method. Hence the JFNK method is a so-called matrix-free variant of the Newton–Raphson method (Knoll and Keyes, 2004).

To apply the JFNK methodology to explicit time-stepping ocean models, one needs the residual $\vec{F}$ of the model and a suitable preconditioner. The approach was used in Nadiga et al. (2006) on the POP model to formulate an implicit time-stepping version of this model, referred to as iPOP. It was shown that much larger time steps could be used than the original POP model and the gain in efficiency was limited by the preconditioner used. The JFNK methodology has also been used to efficiently compute steady state solutions. In Bernsen et al. (2008) this was done for the planetary geostrophic model (Samelson and Vallis, 1995, 1997) for which it is relatively easy to extract the residual from the code. Using the Matrix Renumbering ILU (MRILU) preconditioner (Botta and Wubs, 1999) it was shown that the spin-up time towards steady state is typically reduced by a factor 50. In Khatiwala et al. (2005) the matrix method was introduced and used to accelerate the spin-up of an ocean model.

It is also possible to compute periodic orbits of a seasonally forced ocean model using the JFNK method. In this case the Jacobian matrix is dense rather than sparse causing additional difficulties. For instance it is then no longer feasible to explicitly compute the Jacobian matrix. In Merlis and Khatiwala (2008), the periodic orbits of a seasonally forced single-layer quasi-geostrophic model were computed without using a preconditioner. Periodic solutions have also been computed in biogeochemical models (Khatiwala, 2008; Li and Primeau, 2008, where only the tracer equations are dealt with and the ocean velocity field is prescribed.

Our goal is to bring the JFNK methodology to a widely used ocean model such as MOM4, version 4 of the Modular Ocean Model (Griffies et al., 2004) and use it for the computation of steady states. The challenge here is that in comparison with for instance Bernsen et al. (2008) and Merlis and Khatiwala (2008) the computation of the residual is not straightforward, and in addition more advanced preconditioning techniques are required. In this paper we make the first step by applying the JFNK methodology to the wind-driven ocean circulation, with a fixed density field. In this way, we deal with the momentum, continuity and the free-surface equations and can make use of the preconditioning techniques as developed in de Niet et al. (2007). The case where also the evolution of temperature and salinity equations are treated is a next step which is ongoing work.

In Section 2 we describe the details of the implementation of the JFNK method to obtain steady states in MOM4 and in particular the preconditioner used. In Section 3, results for typical examples are provided to demonstrate the capabilities of the methodology. We present results of first computations of bifurcation diagrams for MOM4 and demonstrate that, even for wind-driven problems with a prescribed density field where the spin-up is relatively fast, we can considerably reduce the spin-up time for MOM4. The results are summarized and discussed in Section 4, where we also outline the steps to generalize the methodology to the full MOM4 model.

2. Method and implementation

The MOM4 ocean model is described in detail in Griffies et al. (2004). It is a primitive equation model that solves the hydrostatic momentum equations, the continuity equation and the equations for temperature and salinity on a so-called Arakawa B-grid. In addition, a free-surface formulation is applied where the sea-surface height is part of the solution. The time discretization in MOM4 consists of an explicit two-level scheme as will be elaborated on below. One of the most important ingredients of the JFNK method is the availability of the residual $\vec{F}(\vec{x})$ and the extraction of this residual in MOM4 is described in Section 2.1. Note that the residual is extracted for all state variables, including temperature and salinity. In Section 2.2, we formulate for the system with prescribed density a preconditioner that we used to speed-up the convergence of the Krylov method that is employed to solve the linear systems.

2.1. Residual of MOM4

It is efficient to determine the residual $\vec{F}$ of MOM4 using the existing MOM4 time-stepping code as much as possible. To show how we computed this residual, we have to explain in some more detail the spatial and temporal discretization in the model.

The Arakawa B-grid is a staggered grid which has two types of cells: U-cells for horizontal velocities and T-cells for sea-surface height and tracers. The corners of each T-cell consist of the centers of four U-cells and both horizontal velocity components are defined on the same grid points. When using the two level time-stepping scheme implemented in MOM4, the grid is also staggered in time. At time-level $\tau - 1/2$, that is at time $t = \Delta t(\tau - 1/2)$, the variables on T-cells are defined, such as temperature, salinity and pressure. At the time-level τ, variables on the U-cells are defined, such as the horizontal velocities. The only variable defined at both time-levels is the sea-surface height.

A simplified description of the two-level time-stepping scheme is given below. At time-level $\tau - 1/2$, the sea-surface height ($\eta_T^{\tau-1/2}$), temperature ($T^{\tau-1/2}$), salinity ($S^{\tau-1/2}$) and vertical grid spacing ($\Delta z_T^{\tau-1/2}$) are given at T-cells. Furthermore, the horizontal velocities ($\vec{u}^{\tau}$), the vertically integrated velocities ($\vec{U}^{\tau}$) and the vertical grid spacing (Δz_u^{τ}) are available at U-cells and at time-level τ. At time-level τ the sea-surface height is given at U-cells (η_u^{τ}) as well as T-cells (η_T^{τ}). To update the variables from $\tau - 1/2$ to $\tau + 1/2$ and from τ to $\tau + 1$, the following procedure is applied:

(1) Update variables defined on time-level $\tau - 1/2$ to time-level $\tau + 1/2$.
   (a) Compute the tendency (i.e. an approximation of the time derivative) $\delta\eta_T$ of the sea-surface height at time-level τ using $\vec{U}^{\tau}$ and calculate the thickness weighted tendencies, δT and δS, for temperature and salinity. This weighting is needed for the conservation of heat and salinity. Note that for the computation of δT and δS we need not only horizontal velocities, which are given, but also vertical velocities, which follow immediately from the horizontal velocities using the continuity equation.The vertical diffusion terms are usually treated implicitly, but optionally it is also possible to treat them explicitly.

(b) Update the sea-surface height to time-level $\tau + 1/2$:

$$\eta_T^{\tau+1/2} \leftarrow \eta_T^{\tau-1/2} + \delta\eta_T$$

and update the vertical grid spacing $\Delta z_T^{\tau+1/2}$ at T-cells.

(c) Now update T and S from $\tau - 1/2$ to $\tau + 1/2$:

$$(T,S)^{\tau+1/2} = \frac{\Delta z_T^{\tau-1/2}(T,S)^{\tau-1/2} + \Delta t(\delta T, \delta S)}{\Delta z_T^{\tau+1/2}}.$$

(2) Update variables defined on time-level τ to time-level $\tau + 1$.

(a) Update the barotropic variables, using sub-time stepping, to obtain $\vec{U}^{\tau+1}$, $\eta_u^{\tau+1}$ and $\eta_T^{\tau+1}$.

(b) Update the vertical grid spacing $\Delta z_u^{\tau+1}$ at U-cells using the sea-surface height obtained in the previous step.

(c) Compute the thickness weighted baroclinic velocity tendencies, say $\delta\vec{u}$. These are obtained from the momentum equations, where the term $\Delta z_u g \nabla\eta$ is omitted from the pressure gradient term.

(d) Compute the velocity field at $\tau + 1$ as follows:

$$\vec{u}^{\tau+1} = \frac{\Delta z_u^{\tau}\vec{u}^{\tau} + \Delta t\delta\vec{u}}{\Delta z_u^{\tau+1}}$$

and correct these velocities such that the vertically integrated velocity matches the vertically integrated velocity obtained from the barotropic sub-time stepping

$$\vec{u}^{\tau+1} \leftarrow \vec{u}^{\tau+1} + \frac{\vec{U}^{\tau+1} - \sum \Delta z_u^{\tau+1}\vec{u}^{\tau+1}}{\sum \Delta z_u^{\tau+1}}$$

where the summation is over all grid points in the same fluid column.

To calculate the residual function $\vec{F}(\vec{x})$ of MOM4, we use a state vector $\vec{x}$ defined as

$$\vec{x} = [\vec{x}_u, \vec{x}_\eta, \vec{x}_T, \vec{x}_S], \tag{5}$$

consisting of the temperature ($\vec{x}_T$), salinity ($\vec{x}_S$) and sea-surface height field ($\vec{x}_\eta$) at T-cells and the horizontal velocity field ($\vec{x}_u$) at U-cells. Assuming that these fields are available we use the following algorithm to compute the residual $\vec{F}$:

(1) Compute variables which depend directly on the state vector, such as the sea-surface height at U-cells, η_u, vertical grid spacing at T and U-cells, Δz_T and Δz_U and vertically integrated velocity field $\vec{U}$. The vertical grid spacing depends directly on the sea-surface height fields while the vertically integrated horizontal velocity depends directly on the vertical grid spacing and the 3D velocity field $\vec{u}$. The sea-surface height at U-cells is obtained by linear interpolation of the sea-surface height at T-cells.

(2) Compute the tendency $\delta\eta_T$ of η_T using $\vec{U}$ and calculate the thickness weighted tendencies, δT and δS, for temperature and salinity.Note that here we use the option to treat vertical diffusion explicitly rather than implicitly.

(3) Compute tendencies for horizontal velocities.

(a) Compute the thickness weighted baroclinic velocity tendencies, $\delta\vec{u}$. These are obtained from the momentum equations, where the term $\Delta z_u g \nabla\eta$ is omitted from the pressure gradient term.

(b) Correct for the omitting part in the pressure gradient term by adjusting the thickness weighted velocity tendency as follows:

$$\delta\vec{u} \leftarrow \delta\vec{u} - \Delta z_u g \nabla\eta.$$

The residual is now given by the vector $\vec{F}(\vec{x}) = [\delta\vec{u}, \delta\eta_T, \delta T, \delta S]$. Note that the residual as calculated here contains thickness weighted tendencies rather than normal tendencies. However, we can use this residual for the calculation of steady states because (2) is still valid.

Steps (2) and (3a) of the residual calculation require most of the coding, but these are the steps for which we can directly reuse code from the time-stepping algorithm. Step (1) is relatively easily implemented because of the modular setup of MOM4. Subroutines for calculating the vertical grid spacing at U and T-cells from the sea-surface height are directly available and so are subroutines for interpolating fields from T-cells to U-cells. Only the vertically integrated velocity field needs to be computed by hand in step (1), but this takes only a few lines of code. For step (3) a subroutine for calculating the omitted term was already available.

There is one issue with the free-surface formulation. Due to volume conservation of the computational domain it holds that the integral of the time derivative of the sea-surface height over the domain (it is only defined on a 2D domain) is zero. By construction this property is retained in the numerical scheme and hence $\eta_T \cdot \vec{v} = 0$, where $\vec{v}$ contains the area of the grid cell on which the corresponding element in η_T is defined. Since the time derivative of η_T is a part of the residual vector $\vec{F}(\vec{x})$ we have that $\vec{F}(\vec{x}) \cdot \vec{w} = 0$, where $\vec{w} = [0, \vec{v}, 0, 0]$ is just $\vec{v}$ extended by zeros for the other components. The resulting Jacobian satisfies

$$J^T\vec{w} = 0$$

and with $\vec{w} \neq 0$ this means that the Jacobian is singular and the JFNK method will break down. To solve this issue we remove one of equations for the sea-surface height from the residual and replace it with the condition

$$\vec{F}(\vec{x}) \cdot \vec{w} = 0. \tag{6}$$

The resulting Jacobian matrix will generically be non-singular and we can apply the JFNK method.

Hence, using this algorithm we can calculate the residual of MOM4 efficiently and reuse most of the original MOM4 code. The same subroutines that are required for time stepping are also useful for the calculation of the residual and only the order in which the subroutines are called slightly changes.

2.2. Preconditioner

A preconditioner is required in order to solve the large sparse linear systems resulting from the Newton–Raphson method. Here we use a right preconditioner, so instead of solving the system

$$J\delta x = b$$

we solve

$$JM^{-1}y = b \tag{7}$$

and obtain δx from $M^{-1}y = \delta x$. The preconditioner M should be chosen such that M^{-1} can be applied efficiently to arbitrary vectors and the Krylov method should have better convergence properties for the matrix JM^{-1} than for J itself.

We now restrict to the case where the density is prescribed in MOM4, so the state vector reduces to

$$\vec{x} = [\vec{x}_u, \vec{x}_\eta]. \tag{8}$$

In this case the Jacobian matrix is of the form

$$J = \begin{bmatrix} A & G \\ D & K \end{bmatrix} \tag{9}$$

where A is the block containing advection and diffusion of momentum and the Coriolis acceleration. The G block contains the gradient of sea-surface height. The D block represents an operator that first

vertically integrates the velocity and then computes the divergence. The K block represents the smoothing operator used by MOM4 to suppress the null-mode present in B-grid ocean models.

In Murphy et al. (1999), it was shown that choosing

$$M = \begin{bmatrix} A & G \\ 0 & S \end{bmatrix}$$

with the Schur complement $S = K - DA^{-1}G$, results in a convergence of the GMRES method within a maximum of two iterations. Computing the Schur complement explicitly is hardly possible since A^{-1} is usually dense rather than sparse. Hence we replace S with an approximation $\widetilde{S} = K - D\widetilde{A}^{-1}G$ where $\widetilde{A}^{-1}$ is a sparse approximation of the inverse of A. Using the method of Barnard and Grote (1999) we compute the matrix $\widetilde{A}^{-1}$, such that it has the same sparsity pattern as A and minimizes

$$\left\| I - A\widetilde{A}^{-1} \right\|_F.$$

with $\|\cdot\|_F$ the Frobenius norm.

Applying the preconditioner M is done by solving the system

$$M \begin{bmatrix} \delta u \\ \delta \eta \end{bmatrix} = \begin{bmatrix} A & G \\ 0 & \widetilde{S} \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \eta \end{bmatrix} = \begin{bmatrix} b_u \\ b_\eta \end{bmatrix} \tag{10}$$

which is comprised of the following two steps

(a) Solve $\widetilde{S}\delta\eta = b_\eta$.
(b) Solve $A\delta u = b_u - G\delta\eta$.

The linear systems in each of these two steps are solved using the GMRES (Frayssé et al., 2003) method (the inner iteration). Note that we don't want to solve these two smaller systems with a very high accuracy every time and therefore to solve the preconditioned system (7) we use a Flexible-GMRES (FGMRES) method (Frayssé et al., 1998) (the outer iteration) which can deal with a variable right preconditioner.

Since the matrix $\widetilde{A}^{-1}$ is sparse, the approximation of the Schur complement $\widetilde{S} = K - D\widetilde{A}^{-1}G$ is sparse as well. An incomplete LU factorization using MRILU (Botta and Wubs, 1999) can easily be computed as a preconditioner for the system in (a) above. For the system in (b) a preconditioner is simply given by $\widetilde{A}^{-1}$.

We note that to compute $\widetilde{A}^{-1}$ and $\widetilde{S}$, the matrices A, G, K and D are required. Assuming the sparsity pattern of the Jacobian matrix J is known (this is obtained from studying the code carefully) we can use Coleman's algorithm (Coleman et al., 329–345) and software (Coleman et al., 1984a) to efficiently compute J and subsequently pick out the blocks A, G, K and D. The algorithm is used to find a partition $C_1, C_2, \ldots, C_p$ of the columns of J such that no pair of columns in the same C_q shares a non-zero element on the same row. Then by performing the matrix-vector product $J\vec{v}$ with $v_i = 0$ if $i \notin\in C_q$ and $v_i = 1$ elsewhere, the columns of J can be obtained. Since for each row of J which contains a non-zero element in column $k \in C_q$ and because no other column in C_q contains a non-zero element in this row, we have that

$$J_{jk} = \sum_{j \in C_q} J_{ij} v_j = \sum_j J_{ij} v_j = (J\vec{v})_j. \tag{11}$$

It is important that Coleman's method is applied to the residual $\vec{F}(\vec{x})$ without the constraint given in (6) on the sea-surface height and then one of the rows of the computed Jacobian matrix is manually replaced with $\vec{w}^T$. If Coleman's method is applied to the residual including the constraint on sea-surface height, then the number of matrix-vector products required to evaluate the Jacobian matrix depends quadratically on the horizontal resolution because the integral condition on sea-surface height depends on the whole sea-surface height field. If it is applied to the residual without the constraint on sea-surface height, then the required number of matrix vector products does not depend at all on the horizontal resolution used.

The vertical velocities are not part of the state vector, and if they are needed they are computed by combining the horizontal velocity field and the continuity equation. The vertical velocity field then depends on vertical columns of velocity points. The residual of the horizontal momentum equations then also depends on vertical columns of velocity points since it contains the vertical velocity. Additionally the residual for sea-surface height contains the vertically averaged horizontal velocity and hence also depends on vertical columns of velocity points. Therefore the number of matrix-vector products required to evaluate the Jacobian matrix depends linearly on the number of layers.

To reduce the number of matrix-vector products for the evaluation of J we apply Coleman's method with a sparsity pattern where the dependencies for the momentum equations are restricted to points in the same layer, the layer above and the layer below (ignoring the dependencies in all the other layers). Using this reduced sparsity pattern we can compute an approximated Jacobian matrix, which resembles the exact Jacobian matrix well enough to use it for the construction of the preconditioner. However, the computation of the preconditioner now consumes much less CPU time, since the approximated Jacobian can be evaluated using fewer matrix–vector products and the resulting matrix A is now much more sparse, resulting in a faster computation of the sparse approximate inverse $\widetilde{A}^{-1}$ as well.

Finally, we would like to note that the preconditioner that we use is not the only possible choice. Since the whole Jacobian is approximated using Coleman's method other preconditioning techniques are relatively easily explored. For instance it is possible to use a sparse direct solver to apply the preconditioner. Alternative preconditioners certainly deserve further investigation, but that is not within the scope of this article.

3. Results

We test the JFNK method for several idealized MOM4 configurations. In all cases, the temperature and salinity fields have prescribed values and we solve for the sea-surface height and velocity field. In Section 3.1, we consider a small basin set-up with a constant density and an equidistant vertical grid, and compare the JFNK method with the original time-stepping method in computing a spin-up steady state. In Section 3.2, this spin-up state is used as a starting point for the computation of a bifurcation diagram with respect to a varying horizontal friction coefficient using MOM4. Finally, in Section 3.3 we consider the spin-up in a realistic size domain under a given stratification using a non-equidistant vertical grid.

3.1. Barotropic spin-up

The domain chosen is a spherical sector with latitudinal range of $40^\circ\text{N} = \theta_{min} \leqslant \theta \leqslant \theta_{max} = 50^\circ\text{N}$, with a longitudinal range of 10° and with a constant depth of 2400 m. The model has 12 layers of equal depth and at the surface a wind stress

$$\tau^\phi = \alpha\tau_0 \cos\left(2\pi \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right) \tag{12a}$$

$$\tau^\theta = 0 \tag{12b}$$

is applied with $\tau_0 = 0.1$ Pa. Here α serves as a so-called 'homotopy' parameter as will become clear below. A uniform density field is prescribed with $\rho = 1058\ \text{kg m}^{-3}$ and we use a horizontal friction of $A_H = A_H^0 = 400\ \text{m}^2\ \text{s}^{-1}$ and a vertical friction given by $A_V = 10^{-3}\ \text{m}^2\ \text{s}^{-1}$.

First we use MOM4 to spin-up the model using the two-level time-stepping method, starting from a state of rest, i.e., $\vec{u} = 0$ and $\eta = 0$, at a horizontal resolution of 20×20 ($0.5^\circ \times 0.5^\circ$) grid points. We use a (baroclinic) time step of $\Delta t = 7200$ s and 40 barotropic sub-timesteps per baroclinic time step and integrate the model forward in time for a period of 5 years. Fig. 1(a) shows the maximum of the barotropic streamfunction (ψ_B) as a function of time which indicates that a steady state is reached after about 5 years of integration. The pattern of the barotropic streamfunction of this steady state is a double-gyre flow as shown in Fig. 1(b).

To check the explicit computation of the residual $\vec{F}(\vec{x}_n)$ as described in Section 2.1, we compare it with the following finite difference approximation (for large n)

$$\vec{F}^*(\vec{x}_n) \equiv \frac{\vec{x}_{n+1} - \vec{x}_n}{\Delta t} \tag{13}$$

with $\vec{x}_n$ and $\vec{x}_{n+1}$ the state vector at two successive time steps in MOM4. We note that in these calculations we use a scaled statevector and residual. For the statevector this scaling is given by

$$\vec{x}'_u = \mathscr{X}_u \vec{x}_u \quad \vec{x}'_\eta = \mathscr{X}_\eta \vec{x}_\eta$$

with $[\vec{x}_u, \vec{x}_\eta]$ and $[\vec{x}'_u, \vec{x}_\eta]$ the scaled and unscaled statevector, respectively. The scaling factors are given by $\mathscr{X}_u = 0.2\ \mathrm{m^2\ s^{-1}}$ and $\mathscr{X}_\eta = 0.5$ m. For the residual a similar scaling is used which is

$$\vec{F}'_u = \mathscr{F}_u \vec{F}_u \quad \vec{F}'_\eta = \mathscr{F}_\eta \vec{F}_\eta$$

with the scaling factors given by $\mathscr{F}_u = 2.0 \cdot 10^{-7}\ \mathrm{m^2\ s^{-2}}$ and $\mathscr{F}_\eta = 2.0 \cdot 10^{-4}\ \mathrm{m\ s^{-1}}$.

Since the approximation (13) is only valid for $\Delta t \to 0$, we performed the time-stepping spin-up run for $\Delta t = 7200$ s, $\Delta t = 3600$ s, $\Delta t = 1800$ s and $\Delta t = 900$ s and calculated the residual by the finite difference approximation, $\vec{F}^*(\vec{x})$, as well as by the implementation described in Section 2.1.

When comparing $\vec{F}(\vec{x}_n)$ to $\vec{F}^*(\vec{x}_n)$ we are mainly interested in the residual for the momentum balance, since here the largest differences are expected. After all, as we explained in Section 2.1, in the computation of $\vec{F}(\vec{x}_n)$ the term $-g\Delta z_u \nabla\eta$ appearing in the momentum balance is evaluated explicitly, while in $\vec{F}^*(\vec{x}_n)$ this term is included implicitly using sub-time stepping.

In Fig. 2(a) the residual of the momentum balance, normalized by the value at $t = 0$, $||\vec{F}(\vec{x})||/||\vec{F}(\vec{x}_0)||$, with $\vec{x}_0$ the statevector at t=0, is given as a function of time. The normalized residual as calculated by (13), $||\vec{F}^*(\vec{x})||/||\vec{F}(\vec{x}_0)||$, is shown only for $\Delta t = 900$ s since the results for other values of Δt are almost identical. In Fig. 2(a), we see that the residual $\vec{F}^*_u(\vec{x})$ approaches zero as time increases, while the norm of the residual $\vec{F}_u(\vec{x})$ stops decreasing after a certain time.

The time stepping in MOM4 is a fractional step like method (Yanenko, 1971) and fits the form

$$\vec{x}_{n+1} = \vec{x}_n + \Delta t(\vec{F}(\vec{x}_n) - \vec{H}(\Delta t, \vec{x}_n))$$

where $\vec{H}(\Delta t, \vec{x}_n) = \mathscr{O}(\Delta t^m)$, with $m > 0$ for a consistent scheme. Hence for a fixed point of the iteration we have that $\vec{F}(\vec{x}) = \vec{H}(\Delta t, \vec{x})) = \mathscr{O}(\Delta t^m)$. For $m = 1$ this is consistent with the results in Fig. 2(a), because a reduction of Δt with a factor of two results in a reduction of the norm of $\vec{F}_u(\vec{x})$ with a factor of two. Note that this also implies that the time-stepping scheme is only first order accurate, due to the Euler-Forward type of method that is used for time stepping the horizontal diffusion and friction terms. We checked the convergence of the velocity field obtained directly from the time-stepping run which confirmed the first-order convergence rate.

Now being confident that our implementation of the residual calculation is correct, we try to use the JFNK method to reach the same steady state solution; here we use the homotopy parameter α in the wind forcing. We start with no forcing, $\alpha = 0$, and know that the no-flow solution, given by $\eta = 0$ and $\vec{u} = 0$, is a steady state. We now use natural parameter continuation in the forcing strength α, i.e., we increase the forcing strength in small steps of $\Delta\alpha = 0.05$, until $\alpha = 1$ is reached and use the JFNK method to obtain the corresponding steady states.

The stopping criterion for Newton's method in all continuation steps was set by $\|\vec{F}(\vec{x})\| \leqslant N\epsilon_N$ with $\epsilon_N = 10^{-8}$ and with N the dimension of $\vec{x}$. It was not needed to use a globalization method in the Newton process to ensure convergence to a solution. For the FGMRES method we used the implementation of Frayssé et al. (1998) and we applied it with a maximum number of 50 iterations and a restart after 25 iterations. The outer FGMRES iteration is stopped when the norm of the relative residual of the preconditioned system has fallen below 10^{-3} while the linear systems with A and $\widetilde{S}$ (Section 2.2) are both solved until the relative residual reached a value of 10^{-4}. If the outer FGMRES solver does not converge within 50 iterations, then the preconditioner M is updated. Convergence of inner GMRES solvers depends on the quality of the preconditioners. It turns out that systems with the matrix A converge quite fast in all cases while for the system with $\widetilde{S}$ the rate of convergence depends on the drop-tolerance that determines the

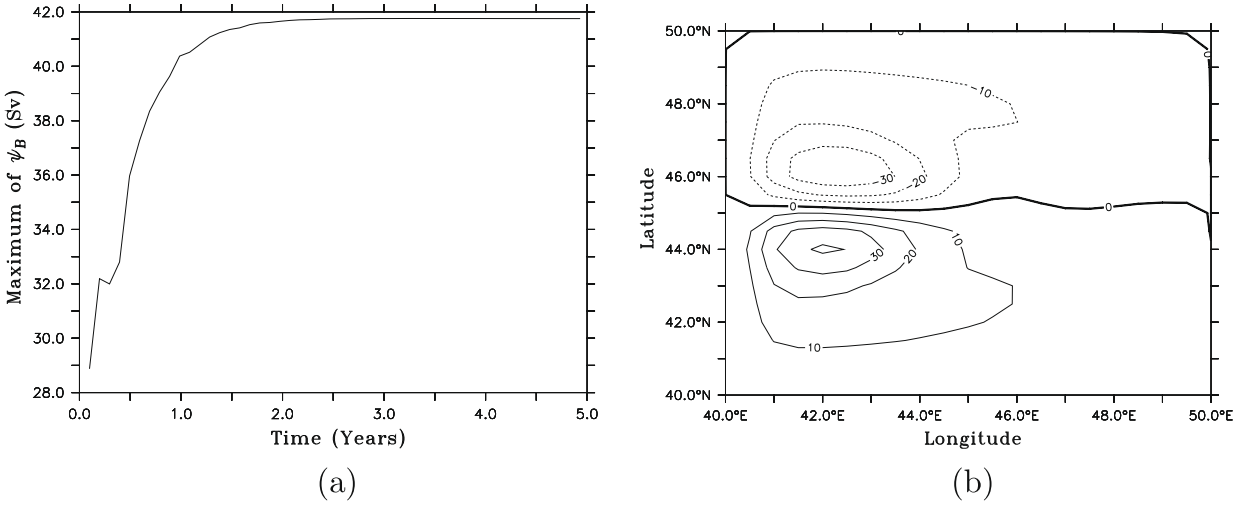

Fig. 1. (a) Maximum of the barotropic streamfunction (ψ_B) as a function of time obtained from the 5-years spin-up run. (b) Pattern of the barotropic streamfunction after 5 years of integration; contour values are in Sverdrup.

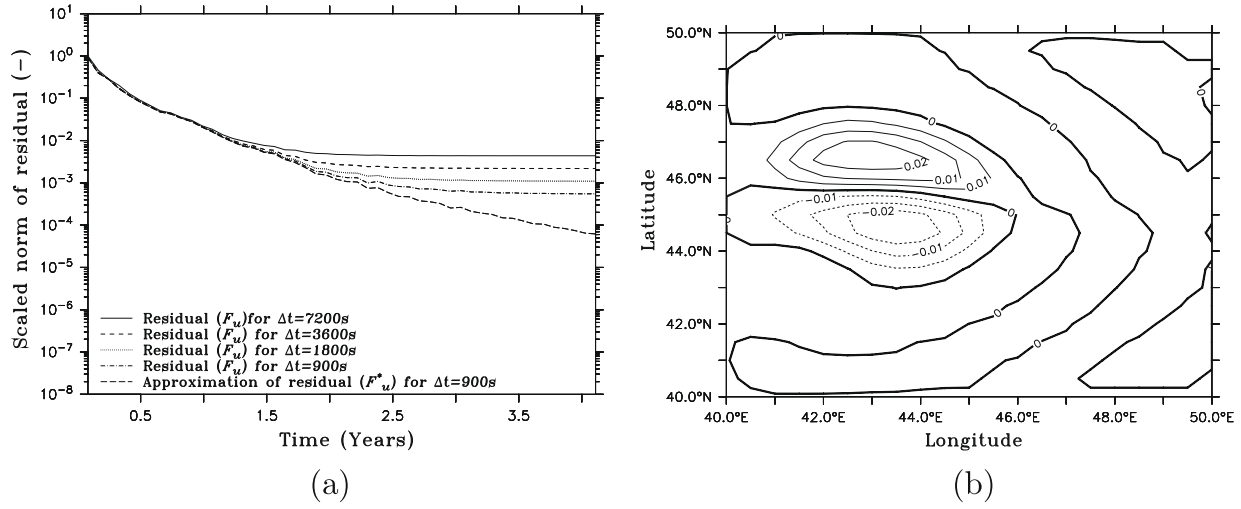

Fig. 2. (a) Scaled norm of the residual of the momentum equations $\vec{F}_u(\vec{x})$ for several time step sizes, Δt, as calculated in Section 2.1 and $\vec{F}^*_u(\vec{x})$ as calculated by (13) for the spin-up run. All curves are normalized with respect to the norm of the residual at $t = 0$. (b) Difference in the barotropic streamfunction between the equilibrium solutions computed with the JFNK method and with the time-stepping method.

fill-in in the LU factorization. This drop-tolerance was chosen low enough to have convergence within 40 GMRES iterations in all examples considered in this article. For the GMRES method the implementation of Frayssé et al. (2003) was used. In Fig. 2(b) it is shown that the difference between the barotropic streamfunction of the solution of the time-stepper after 5 years and the JFNK method are relatively small, with a maximum value of $2 \cdot 10^{-2}$ Sv and a corresponding relative error of $1 \cdot 10^{-3}$.

We now increase the horizontal resolution to 40×40 and 80×80 grid points and use the JFNK method and the time-stepping spin-up run to obtain the equilibrium solutions. In Table 1 the CPU-time and the maximum of the barotropic streamfunction is given for all three resolutions and for both the time-stepping method and the JFNK method. On all resolutions the time-stepper and JFNK method converge to the same solution, while the computational cost of both methods depends on the desired accuracy. For the JFNK method the accuracy of the approximation of the equilibrium solution is set by the desired accuracy of the Newton process, ϵ_N, while for the time-stepper the accuracy of the equilibrium solutions is determined by the integration period; a longer integration period results in a more accurate approximation of the equilibrium solution. When we compare the CPU time of the JFNK method to a spin-up run of 5 years we see that the JFNK method is much faster, a factor 5–15 depending on resolution. If a less accurate solution is required, then a shorter integration period could be used, but the JFNK method remains faster, unless an unreasonably short spin-up period of approximately 9 months is used. At the lowest resolution of 20×20 grid points, such a short spin-up would result in an error in the barotropic streamfunction of at least 4 Sv.

Table 1
Comparison of CPU time for the computation of the equilibrium solution obtained with the JFNK method and the solution after 5 years of time stepping in MOM4. The CPU time for the JFNK method includes the time required for computing and applying the preconditioner.

Resolution	N	CPU Time		Maximum of ψ_B		Speed-up
		JFNK	5 years of time-stepping	JFNK	5 years of time-stepping	
20×20	9064	$6.7 \cdot 10^1$ s	$4.6 \cdot 10^2$ s	41.7538 Sv	41.7552 Sv	6.9
40×40	38104	$3.2 \cdot 10^2$ s	$3.4 \cdot 10^3$ s	43.7142 Sv	43.7132 Sv	10.6
80×80	156184	$1.7 \cdot 10^3$ s	$2.5 \cdot 10^4$ s	43.6770 Sv	43.6701 Sv	14.1

3.2. Bifurcation analysis

For a barotropic quasi-geostrophic model of the double-gyre wind-driven ocean circulation, the bifurcation diagram for changing the horizontal friction coefficient A_H was determined in Cessi et al. (1995) and Dijkstra and Katsman (1997). For large values of A_H, a unique anti-symmetric steady barotropic streamfunction solution exists, but when A_H is decreased at some point a symmetry-breaking pitchfork bifurcation is encountered. The anti-symmetric steady state becomes unstable and two additional asymmetric stable steady states appear, the so-called jet-up and jet-down solution. In de Niet et al. (2007), a similar bifurcation diagram was found for the primitive equation model THCM, but because a spherical sector instead of a β-plane was used, the pitchfork bifurcation becomes imperfect. For MOM4 we also expect an imperfect pitchfork, since the only difference between THCM and MOM4, in this idealized set-up, is in the treatment of the ocean-atmosphere surface; MOM4 uses a free-surface while THCM uses a rigid-lid approximation.

Starting from the steady solution as calculated in the previous subsection on a horizontal 80×80 grid, we now determine how the steady states of MOM4 depend on A_H. Thereto we introduce another homotopy parameter λ defined by $A_H = \lambda A_H^0$, with $A_H^0 = 400$ m^2 s^{-1}. We first use natural parameter continuation in λ and decrease the parameter λ from $\lambda = 1$ in steps of $\Delta\lambda = 0.05$ to $\lambda = 0.5$ ($A_H = 200$ m^2 s^{-1}) and find the asymmetric jet-down solution indicated by point B in Fig. 3. The spatial pattern of the barotropic streamfunction for this solution is shown in Fig. 4(b).

In quasi-geostrophic models the jet-up and the jet-down solution are related by a reflection symmetry with respect to the mid-axis of the basin (here, along the line $\theta = 45^\circ$). Because of the spherical geometry, the symmetry is slightly perturbed in MOM4 and hence an imperfect pitchfork bifurcation is expected, and the jet-up and jet-down solution are only approximately symmetry related. To find the jet-up solution from the jet-down solution at location B, we can use the approximate symmetry in a residual continuation approach (Gruais et al., 2005). Thereto we replace the residual $\vec{F}(\vec{x})$ by

$$\vec{G}(\vec{x};\mu) = \vec{F}(\vec{x}) - (1-\mu)\vec{F}(\vec{x}_0) \tag{14}$$

with $\vec{x}_0$ the vector corresponding to the state given by

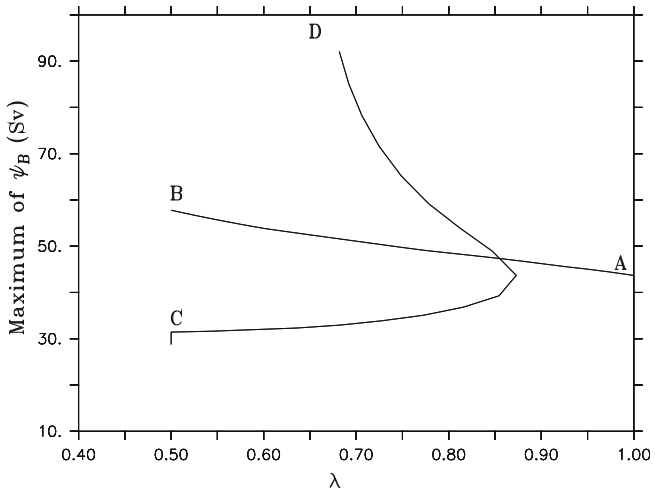

Fig. 3. Maximum of the barotropic streamfunction as a function of $\lambda = A_H/A_H^0$, with $A_H^0 = 400\mathrm{m}^2\mathrm{s}^{-1}$. The solution for the barotropic streamfunction at the points A, B, C and D are shown in Fig. 4(a–d), respectively.

$$u(\phi,\theta,z) = v_B(\phi,\theta_{max}-\theta,z) \tag{15a}$$
$$v(\phi,\theta,z) = -v_B(\phi,\theta_{max}-\theta,z) \tag{15b}$$
$$\eta(\phi,\theta) = \eta_B(\phi,\theta_{max}-\theta) \tag{15c}$$

with u_B, v_B and η_B the horizontal velocities and sea-surface height of the solution corresponding to point B in Fig. 3 (θ_{max} is the northern boundary of the basin). We then consider non-linear systems of the form

$$\vec{G}(\vec{x};\mu) = 0. \tag{16}$$

If $\mu = 1$ then $\vec{G}(\vec{x};\mu) \equiv \vec{F}(\vec{x})$ and solutions of (16) correspond to steady states of MOM4, while for $\mu = 0$ we already have a solution of (16) which is given by $\vec{x}_0$ (note that $\vec{F}(\vec{x}_0) \neq 0$)). Using natural parameter continuation in μ, starting from $\mu = 0$ and increasing μ in steps $\Delta\mu = 0.1$ to $\mu = 1$ we find the jet-up solution corresponding to point C in Fig. 3. The spatial pattern of the barotropic streamfunction of this solution is shown in Fig. 4(c).

Starting from the jet-up solution at point C, we increase the horizontal friction coefficient C again. Because we now expect to find a saddle-node bifurcation (as part of the imperfect pitchfork), we use pseudo-arclength continuation (Keller, 1977) instead of natural parameter continuation. In Fig. 3, we see that indeed a saddle-node bifurcation is found for $\lambda = A_H/A_H^0 \approx 0.87$; for $\lambda < 0.87$ three steady states exist, while for $\lambda > 0.87$ there is only one steady state. Following the branch further in λ, we end up in point D in Fig. 3 which corresponds to an almost anti-symmetric solution of which the pattern of the barotropic streamfunction is plotted in Fig. 4(d). Note that the intersection of the curves $A-B$ and $C-D$ does not correspond to a bifurcation point, but only means that the maximum of the barotropic streamfunction is equal for two

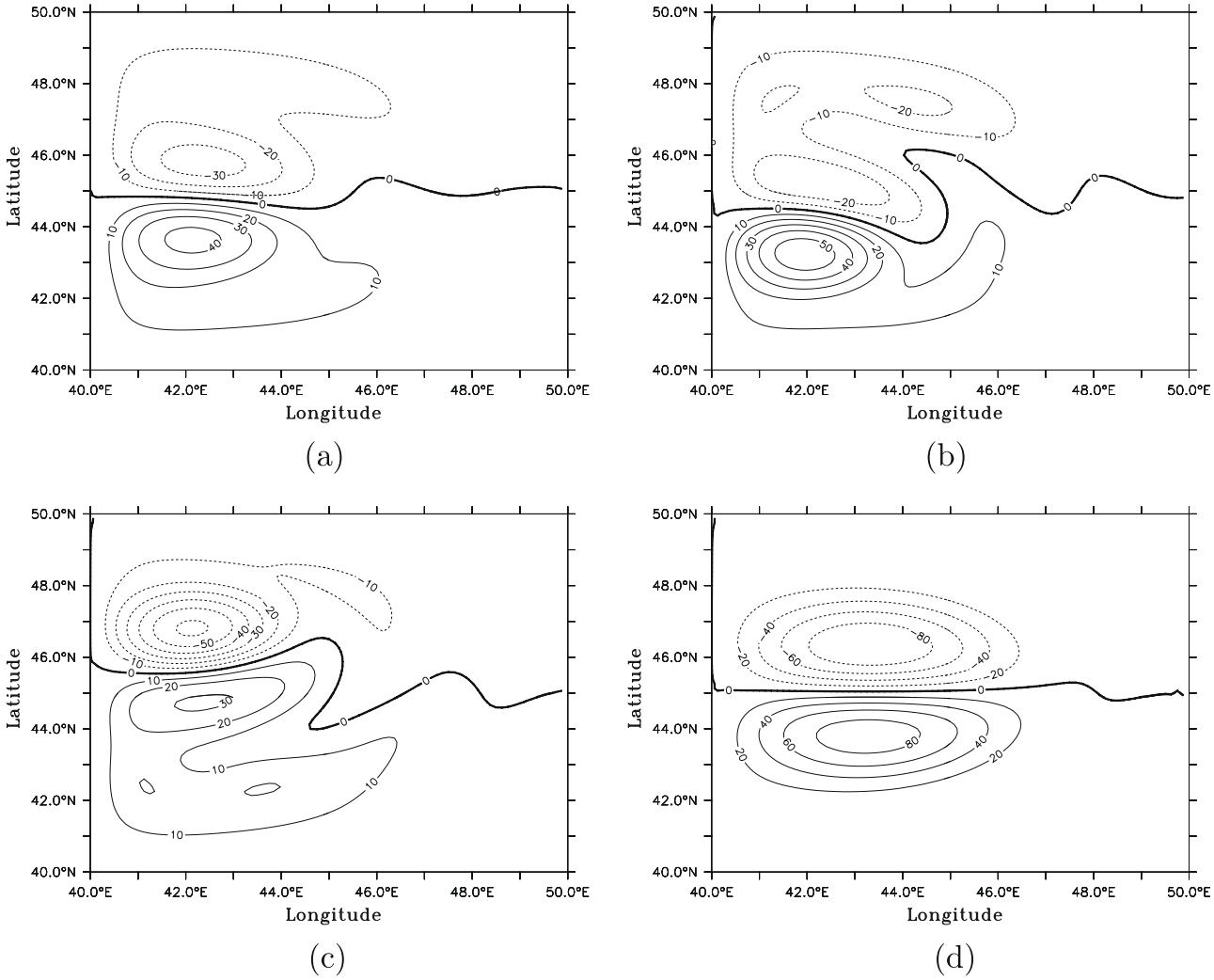

Fig. 4. Barotropic streamfunction of the solutions corresponding to the points labeled a, b, c and d in Fig. 3, respectively; contour levels are in Sverdrup.

otherwise very different solutions along these two curves. We performed a short time integration from point D in Fig. 3 to verify that the steady state is unstable, which is indeed the case. These unstable steady states cannot be found by MOM4 in using it's original time-stepping method only.

The computation of this bifurcation diagram adequately demonstrates the ability to perform bifurcation analysis on MOM4 using the JFNK method.

3.3. Baroclinic spin-up

We now consider a domain consisting of a spherical section with a latitudinal range of $10^\circ\text{N} = \theta_{min} \leqslant \theta \leqslant \theta_{max} = 74^\circ\text{N}$ and a longitudinal range of 64° at a resolution of $4^\circ \times 4^\circ$, $2^\circ \times 2^\circ$ and $1^\circ \times 1^\circ$. The depth of the basin is constant 5500 m and we use 16 layers whose thickness varies in the vertical, ranging from 900 m at depth to 25 m at the surface.

An idealized wind stress of the form (12a) with $\alpha = 1$ ($\tau_0 = 0.1$ Pa) is prescribed at the surface. Surface temperature and surface salinity are restored to

$$T = \Delta T \cos\left(\pi \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right) \tag{17a}$$

$$S = \Delta S \cos\left(\pi \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}\right) \tag{17b}$$

with $\Delta T = 20\,^\circ\text{C}$ and $\Delta S = 1$ psu. For the vertical friction we take $A_V = 10^{-3}\ \text{m}^2\ \text{s}^{-1}$ while the horizontal friction coefficient is given by $A_H = 2.5 \cdot 10^5\ \text{m}^2\ \text{s}^{-1}$, $A_H = 1.25 \cdot 10^5\ \text{m}^2\ \text{s}^{-1}$ and $A_H = 6.25 \cdot 10^4\ \text{m}^2\ \text{s}^{-1}$ for a resolution of $4^\circ \times 4^\circ$, $2^\circ \times 2^\circ$ and $1^\circ \times 1^\circ$, respectively. In MOM4, we first compute a steady state using a 1500-years time-stepping simulation for each of the three resolutions. The density field of this steady state (with respect to a constant reference density) is shown in Fig. 5(a) for a resolution of $1^\circ \times 1^\circ$. A stably stratified solution is obtained with a pycnocline at a depth of 1000 m and a very weak stratification north of 65°N. Similar results are found for a resolution of $2^\circ \times 2^\circ$ and $4^\circ \times 4^\circ$. In any subsequent computation below, the density field is prescribed to the one found by the 1500-years time-stepping simulation.

For each of the three horizontal resolutions we now perform a time-stepping run, from no-flow initial conditions given by $\vec{u} = 0$ and $\eta = 0$ with time steps $\Delta t = 86400$ s, $\Delta t = 43200$ s and $\Delta t = 21600$ s, respectively. Because the density field is prescribed an equilibrium solution is reached after an integration period of only 10 years. The meridional overturning streamfunction (ψ_M) corresponding to this equilibrium solution is given in Fig. 5(b) for a resolution of $1^\circ \times 1^\circ$. With sinking only occurring north of 65°N and a strength of 22 Sv this overturning streamfunction is very similar to those obtained in low-resolution ocean-only models, such as THCM (de Niet et al., 2007).

We now apply the JFNK method, starting from a no-flow initial condition and setting the wind forcing to full strength immediately. Using a stopping criterion of $\epsilon_N = 10^{-12}$ for the Newton–Raphson method, and stopping criteria for GMRES and FGMRES the same as in Section 3.1, the JFNK method converged for all three resolutions.

In Table 2 the convergence history of the JFNK method for all three resolutions is shown. We don't see quadratic convergence of Newton's method because we use an inexact Newton method. Hence we do not solve the system $J\delta\vec{x} = \vec{b}$ exactly, but instead we find a $\delta\vec{x}$ such that it satisfies $||J\delta\vec{x} - \vec{b}|| < 10^{-3}||\vec{b}||$. In particular for the last iterations at the highest resolution it can be seen that the norm of the residual drops with a factor of approximately 10^{-3} during each Newton iteration.

At all resolutions we need 144 matrix-vector products to evaluate J using Coleman's method. As can be seen from Table 2 the number of matrix–vector products used for the GMRES iterations is much smaller and hence it is expected constructing the preconditioner is one of the most costly parts of the algorithm. In Table 3 the relative cost for several parts of the algorithm are given as a percentage of the total CPU time. Clearly constructing the preconditioner takes most of the time, and indeed in particular the computation of J takes a lot of time. The amount of time spend in applying the preconditioner increases for higher resolutions, which is consistent with the increasing number of GMRES iterations for higher resolutions.

In Fig. 6 the maximum value of the difference between the meridional overturning streamfunction as computed by the JFNK method and the time-stepping method is plotted as a function of the scaled CPU time, γ, consumed by the time-stepper. The scaled CPU time is obtained by dividing the actual CPU time of the time-stepping method by the total amount of CPU time to find an equilibrium solution using JFNK method. Hence the scaling factor is different for the three resolutions and always such that at $\gamma = 1$ the CPU time of the time-stepper is equal to that of the JFNK method. From Fig. 6 we conclude that the time-stepper and JFNK method approach the same steady state.

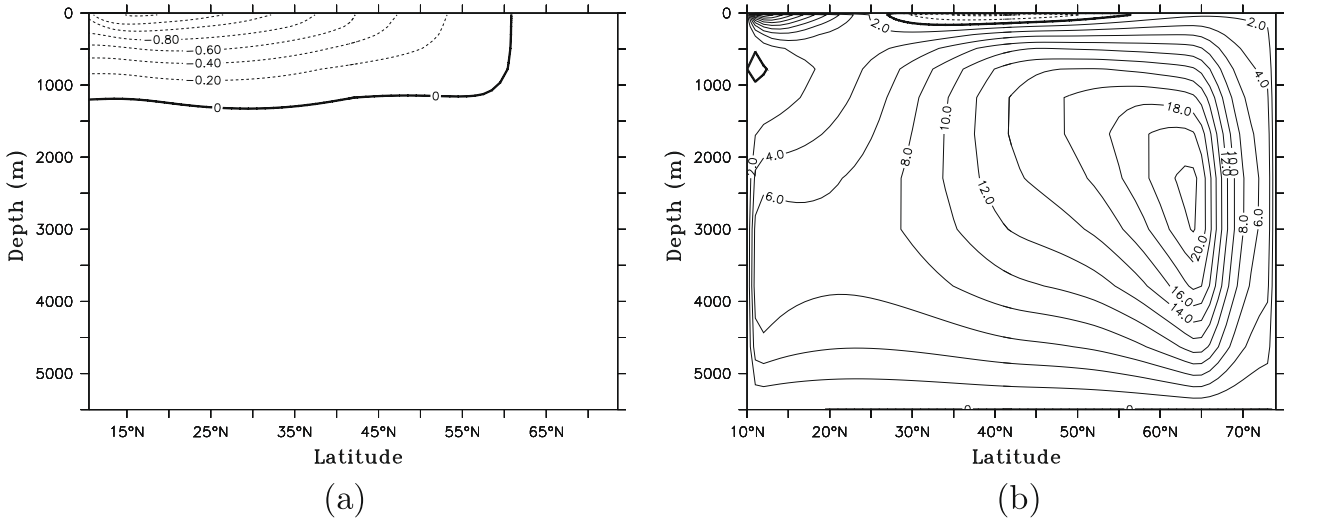

Fig. 5. (a) Zonally averaged density distribution (deviation from a constant reference density) as computed by the simulation under restoring conditions with MOM4 at a $1^\circ \times 1^\circ$ horizontal resolution. (b) Meridional overturning streamfunction of the equilibrium solution.

Table 2
The convergence history of the JFNK method for several resolutions. For each Newton iteration we give the norm of the residual ($||\vec{F}(\vec{x})||/N$) at the end of that iteration and the number of GMRES iterations used in solving the linear system corresponding to this Newton step.

Resolution	16×16 ($N = 7456$)		32×32 ($N = 31776$)		64×64 ($N = 131104$)	
Newton iteration	$\|\|\vec{F}(\vec{x})\|\|/N$	#its	$\|\|\vec{F}(\vec{x})\|\|/N$	#its	$\|\|\vec{F}(\vec{x})\|\|/N$	#its
1	$2.6 \cdot 10^{-2}$	1	$2.6 \cdot 10^{-2}$	1	$2.3 \cdot 10^{-2}$	1
2	$3.7 \cdot 10^{-2}$	26	$3.3 \cdot 10^{-2}$	26	$2.8 \cdot 10^{-2}$	26
3	$3.7 \cdot 10^{-6}$	3	$2.7 \cdot 10^{-5}$	3	$7.9 \cdot 10^{-5}$	5
4	$3.6 \cdot 10^{-9}$	2	$1.3 \cdot 10^{-8}$	4	$2.1 \cdot 10^{-7}$	10
5	$2.8 \cdot 10^{-13}$	3	$1.7 \cdot 10^{-12}$	4	$1.1 \cdot 10^{-10}$	10
6	–	–	$4.2 \cdot 10^{-14}$	4	$8.1 \cdot 10^{-14}$	12

Table 3
CPU time spent in computing the matrix J, using Coleman's method, computation of the sparse approximate inverse $\tilde{A}^{-1}$, computing the MRILU decomposition and application of the preconditioner.

Resolution	16×16 ($N = 7456$) (%)	32×32 ($N = 31776$) (%)	64×64 ($N = 131104$) (%)
Computation of J (Coleman's method)	48	47	37
Computation of $\tilde{A}^{-1}$	12	12	12
MRILU decomposition of $\tilde{S}$	0	0	1
Application of preconditioner	9	13	28
Other	31	28	22

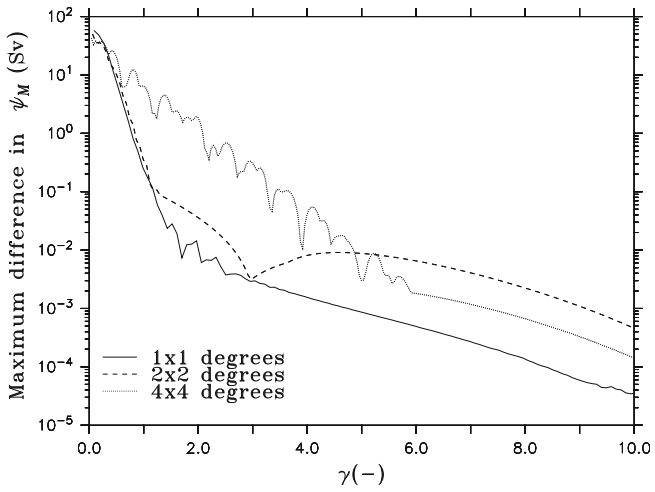

Fig. 6. Maximum difference in the meridional overturning streamfunction (ψ_M) between the solution obtained by the JFNK method and the one obtained by the time stepper. On the horizontal axis the scaled CPU time, γ, consumed by the time stepper is given. A value of $\gamma = 1$ corresponds to the amount of CPU time used by the JFNK method at the same resolution.

We now consider the speed-up obtained by using the JFNK method. Suppose that we want to calculate the meridional overturning streamfunction at an accuracy of 10^{-3} Sv and at a resolution of $4^\circ \times 4^\circ$. The difference between ψ_M corresponding to the equilibrium solution found by the JFNK method and ψ_M found by the time-stepper is a good approximation of the accuracy that the time-stepping method has reached. We find that the time-stepping method reaches the desired accuracy for $\gamma \approx 7.5$ and therefore the JFNK method was approximately 7.5 times faster in finding the equilibrium solution (at a much higher accuracy). Hence, the scaled CPU time, γ, can be interpreted as the speed-up of the JFNK method compared to the time-stepping method. If a very high accuracy is desired then the JFNK method is up to 10 times faster while for low accuracies the time-stepping method is competitive with the JFNK method. At a resolution of $4^\circ \times 4^\circ$ the maximum error in the meridional overturning streamfunction after an amount of CPU time equal to the total CPU time consumed by the JFNK method (i.e. at $\gamma = 1$) is still a very large value of 6.4 Sv while for resolutions of $2^\circ \times 2^\circ$ and $1^\circ \times 1^\circ$ this error has relatively small values of $3.6 \cdot 10^{-1}$ Sv and $2.6 \cdot 10^{-1}$ Sv, respectively.

4. Summary and discussion

We have shown that efficient steady-state computation of MOM4 is possible for a restricted set-up where the density field is prescribed. For the barotropic double-gyre flow in a small basin we presented, for the first time for MOM4, bifurcation diagrams where steady states are determined versus the horizontal friction coefficient A_H. Pseudo-arclength as well as residual continuation methods could be used to compute a regime of multiply steady states with a jet-up and jet-down solution, which exists when the horizontal friction is small enough. Also an unstable steady state was obtained, which is impossible to do with MOM4 with the original time-stepping method.

We also compared the JFNK method with the timestepping method for a baroclinic spin-up where the density field was prescribed but not uniform. Here we found that the JFNK method is considerably faster if a highly accurate solution is required. If a less accurate solution is satisfactory then the two methods are approximately equally fast.

The computation of the steady states was possible by successful application of the JFNK method. One crucial step is the efficient computation of the residual function $\vec{F}$ of MOM4 as presented in Section 2.1. Another approach would be to approximate the residual as follows

$$\vec{F}(\vec{x}) \approx \frac{\vec{Q}(\vec{x}, \Delta t) - \vec{x}}{\Delta t} \tag{18}$$

with $\vec{Q}(\vec{x}, \Delta t)$ the result of one baroclinic time step. This approach has the main advantage that it uses the time-stepping routine immediately, but there are also disadvantages. The dependencies of the Jacobian matrix become much more complicated, not only due to the barotropic time stepping, but also because updated temperatures are used in the calculation of the new horizontal velocity field. Implementation of the finite difference formulation (18) is not that much easier than calculating the residual directly, since one still has to determine what fields are required at the beginning of a time step and the relation with the state variables. We have therefore not pursued this approach further.

Another crucial step in the application of the JFNK method is the construction of the preconditioner to solve the linear systems which arise through the application of the Newton–Raphson iteration. Here, we made efficient use of Coleman's algorithm to compute an approximation of the Jacobian matrix and a variant of the Murphy et al. (1999) preconditioner. Combined with FGMRES, this method converges fast for a range of parameters and ocean-model resolutions.

In all spin-up (barotropic and baroclinic) cases considered, the JFNK method gave a considerable speed-up compared to the time stepping to steady state. Here we must note that we only solved the momentum equations and sea-surface height equations, and hence the spin-up time-scale is on the order of several years. When temperature and salinity are added to the system then the potential gains become much larger because the spin-up time then is on the order of thousands of years. However, a speed-up will only be

achieved if we can successfully extend the preconditioner used in this paper to the full system of equations. The linear systems that we need to solve for the full MOM4 equations are of the form

$$\begin{bmatrix} J & B \\ C & E \end{bmatrix} \begin{bmatrix} \vec{x}_{u\eta} \\ \vec{x}_{TS} \end{bmatrix} = \begin{bmatrix} \vec{b}_{u\eta} \\ \vec{b}_{TS} \end{bmatrix}$$

with J the Jacobian given by (9), B the buoyancy forcing in the momentum equations, C and E the dependency of the tracer equations on horizontal velocity, surface height and temperature and salinity, respectively. The state vector now consists of a part $\vec{x}_{u\eta}$ containing the horizontal velocities and sea-surface height and a part containing temperature and salinity $\vec{x}_{TS}$. One choice of preconditioner would be to use a block Gauss-Seidel preconditioner where the block B in the Jacobian of the full system is neglected in the preconditioner. Applying the preconditioner then requires solving the following block triangular system

$$\begin{bmatrix} J & 0 \\ C & E \end{bmatrix} \begin{bmatrix} \vec{x}_{u\eta} \\ \vec{x}_{TS} \end{bmatrix} = \begin{bmatrix} \vec{b}_{u\eta} \\ \vec{b}_{TS} \end{bmatrix}$$

which can easily be done in the following two steps

(a) Solve $J\vec{x}_{u\eta} = \vec{b}_{u\eta}$ for $\vec{x}_{u\eta}$.
(b) Solve $E\vec{x}_{TS} = \vec{b}_{TS} - C\vec{x}_{u\eta}$ for $\vec{x}_{TS}$.

The preconditioner discussed in this paper then can be applied to step (a) of course. We note that this block Gauss–Seidel preconditioner is in fact very similar to the block preconditioner for THCM as described in de Niet et al. (2007). At the moment the THCM preconditioner is reimplemented using solvers from Trilinos Heroux et al. (2003) for the subsystems. This new version can be run on distributed memory multiprocessor systems with high speed interconnections. Current experiments show a speed-up of a factor 20 when 32 processors are used. For solving the full MOM4 equations we plan to make use of the implementation of this preconditioner.

When applying the JFNK method to other models it is important to realize that it is not always possible to express the model as (1) since algebraic constraints, such as the continuity equation, do not contain time derivatives. However, this is easily dealt with by replacing (1) with $M\mathrm{d}\vec{x}/\mathrm{d}t = \vec{F}(\vec{x}, \lambda)$ where M is a diagonal matrix having the value of one at the diagonal elements for the prognostic equations, including a time-derivative, and a value of zero for algebraic constraints. Steady states of the system now still have to satisfy (2).

Although in this article we used the JFNK method to compute steady states only, it can also be applied to transient runs using a fully implicit time-stepping scheme. Much larger time step can then be used, possibly resulting in a reduction of CPU time. We expect that the systems resulting from these implicit time-stepping schemes will be easier to solve than the full steady system, since it is generally better conditioned due to an increase of the diagonal values of the Jacobian matrix.

For the computation of orbits resulting from a seasonally forced ocean model using JFNK methods (Khatiwala, 2008) the residual, $\vec{R}(\vec{x})$, is usually defined by a time integration over the forcing period,

$$\vec{R}(\vec{x}) = \vec{x} - \vec{Q}^n(\vec{x}, \Delta t)$$

with $\vec{Q}^n(\vec{x}, \Delta t)$ the result of n successive baroclinic time steps and $n\Delta t$ equals the period of the forcing. If a forcing with a high time resolution is prescribed then a relatively small timestep is required for an accurate solution and implicit timestepping is probably not very desirable. On the other hand, if we use a coarse time-resolution forcing, for instance monthly averaged fields, then we can possibly benefit from the larger timesteps that implicit schemes allow and it is in this case that the computation of seasonally forced periodic orbits can possibly benefit from an implicit timestepping scheme.

In summary, we think that with our approach we finally will be able to efficiently compute steady states and seasonal cycles of MOM4. This will open the door for detailed studies of parameter sensitivity of ocean models and will lead to less computational effort to compute spin-up equilibria with these models.

References

Barnard, S.T., Grote, M.J., 1999. A block version of the spai preconditioner. In: Proc. of the 9th SIAM conference on Parallel Processing for Scientific Computing.

Bernsen, E., Dijkstra, H.A., Wubs, F.W., 2008. A method to reduce the spin-up time of ocean models. Ocean Modell. 20, 380–392.

Botta, E.F.F., Wubs, F.W., 1999. Matrix renumbering ILU: an effective algebraic multilevel ILU preconditioner for sparse matrices. SIAM J. Matrix Anal. Appl. 20 (4), 1007–1026.

Cessi, P., Ierley, G., 1995. Symmetry-breaking multiple equilibria in quasigeostrophic, wind-driven flows. J. Phys. Oceanogr. 25, 1196–1205.

Coleman, T.F., Garbow, B.S., Moré, J.J., 1984a. Algorithm 618: Fortran subroutines for estimating sparse Jacobian matrices. AMC Trans. Math. Software 10 (3), 346–347.

Coleman, T.F., Garbow, B.S., Moré, J.J., 1984b. Software for estimating sparse Jacobian matrices. ACM Trans. Math. Software 10 (3), 329–345.

de Niet, A.C., Wubs, F.W., Terwisscha van Scheltinga, A.D., Dijkstra, H.A., 2007. A tailored solver for the bifurcation analysis of ocean–climate models. J. Comput. Phys. 227, 654–679.

Dijkstra, H.A., Katsman, C.A., 1997. Temporal variability of the wind-driven quasi-geostrophic double gyre ocean circulation: basic bifurcation diagrams. Geophys. Astrophys. Fluid Dyn. 85, 195–232.

Dijkstra, H.A., Weijer, W., 2005. Stability of the global ocean circulation: basic bifurcation diagrams. J. Phys. Oceanogr. 35, 933–948.

Frayssé, V., Giraud, L., Gratton, S., 1998. A Set of Flexible-GMRES Routines for Real and Complex Arithmetics. CERFACS.

Frayssé, V., Giraud, L., Gratton, S., Langou, J., 2003. A Set of GMRES Routines for Real and Complex Arithmetics on High Performance Computers. CERFACS. Available from: http://www.cerfacs.fr/algor/reports/2003/TR_PA_03_03.pdf.

Griffies, S., Harrison, M., Pacanowski, R., Rosati, A., 2004. A Technical Guide to MOM4. NOAA/Geophysical Fluid Dynamics Laboratory. Available from: http://www.gfdl.noaa.gov/fms.

Gruais, I., Cousin-Rittemard, N., Dijkstra, H.A., 2005. A priori estimations of a global homotopy residue continuation method. Num. Funct. Anal. Optimiz. 26, 507–521.

Heroux, M., Bartlett, R., Hoekstra, V.H.R., Hu, J., Kolda, T., Lehoucq, R., Long, K., Pawlowski, R., Phipps, E., Salinger, A., Thornquist, H., Tuminaro, R., Willenbring, J., Williams, A., 2003. An Overview of Trilinos. Tech. Rep. SAND2003-2927, Sandia National Laboratories.

Keller, H.B., 1977. Numerical solution of bifurcation and nonlinear eigenvalue problems. In: Rabinowitz, P. (Ed.), Applications of Bifurcation Theory. Academic Press, New York, USA.

Khatiwala, S., 2008. Fast spin up of ocean biogeochemical models using matrix-free Newton–Krylov. Ocean Modell. 23, 121–129.

Khatiwala, S., Visbeck, M., Cane, M.A., 2005. Accelerated simulation of passive tracers in ocean circulation models. Ocean Modell. 9, 51–69.

Knoll, D., Keyes, D., 2004. Jacobian–free Newton–Krylov methods: a survey of approaches and applications. J. Comput. Phys. 193, 357–397.

Li, X., Primeau, F., 2008. A fast Newton–Krylov solver for seasonally varying global ocean biogeochemistry models. Ocean Modell. 23, 13–20.

Merlis, T.M., Khatiwala, S., 2008. Fast dynamical spin-up of ocean general circulation models using Newton–Krylov methods. Ocean Modell. 21, 97–105.

Murphy, M.F., Golub, G.H., Wathen, A.J., 1999. A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comp., 21.

Nadiga, B.T., Luce, B., 2001. Global bifurcation of Shilñikov type in a double-gyre model. J. Phys. Oceanogr. 31, 2669–2690.

Nadiga, B.T., Taylor, M., Lorenz, J., 2006. Ocean modelling for climate studies: eliminating short time scales in long-term, high-resolution studies of ocean circulation. Math. Comput. Modell. 44 (9–10), 870–886.

Primeau, F.W., 2002. Multiple equilibria and low-frequency variability of the wind-driven ocean circulation. J. Phys. Oceanogr. 32, 2236–2256.

Saad, Y., 1996. Iterative Methods for Sparse Linear Systems. PWS Publishing Company.

Samelson, R., Vallis, G.K., 1997. A simple friction and diffusion scheme for planetary geostrophic basin models. J. Phys. Oceanogr. 27, 186–194.

Samelson, R.M., Vallis, G.K., 1995. Planetary-Geostrophic Ocean Model: A User's Guide. Available from: http://www-po.coas.oregonstate.edu/homes/rms/ocean/.

Simonnet, E., Ghil, M., Dijkstra, H.A., 2005. Homoclinic bifurcations of barotropic qg double-gyre flows. J. Mar. Res. 63, 931–956.
Speich, S., Dijkstra, H.A., Ghil, M., 1995. Successive bifurcations of a shallow-water model with applications to the wind driven circulation. Nonlinear Proc. Geophys. 2, 241–268.
te Raa, L.A., Dijkstra, H.A., 2002. Instability of the thermohaline ocean circulation on interdecadal time scales. J. Phys. Oceanogr. 32, 138–160.
Thies, J., 2008. A parallel version of the fully implicit model THCM. Presented at the 8th. World Congress on Computational Mechanics (WCCM8) and the 5th. In: European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008), Venice (Italy), 30 June–4 July 2008.
Yanenko, N.N., 1971. The Method of Fractional Steps. Springer-Verlag www.springer.de .